G-RCA: A Generic Root Cause Analysis Platform for Service Quality Management in Large IP Networks

He Yan, *Member, IEEE*, Lee Breslau, Zihui Ge, *Member, IEEE*, Dan Massey, *Senior Member, IEEE*, Dan Pei, *Senior Member, IEEE, ACM*, and Jennifer Yates, *Member, IEEE*

***Abstract*—An increasingly diverse set of applications, such as Internet games, streaming videos, e-commerce, online banking, and even mission-critical emergency call services, all relies on IP networks. In such an environment, best-effort service is no longer acceptable. This requires a transformation in network management from detecting and replacing individual faulty network elements to managing the end-to-end service quality as a whole. In this paper, we describe the design and development of a Generic Root Cause Analysis platform (G-RCA) for service quality management (SQM) in large IP networks. G-RCA contains a comprehensive service dependency model that incorporates topological and cross-layer relationships, protocol interactions, and control plane dependencies. G-RCA abstracts the root cause analysis process into signature identification for symptom and diagnostic events, temporal and spatial event correlation, and reasoning and inference logic. G-RCA provides a flexible rule specification language that allows operators to quickly customize G-RCA and provide different root cause analysis tools as new problems need to be investigated. G-RCA is also integrated with data trending, manual data exploration, and statistical correlation mining capabilities. G-RCA has proven to be a highly effective SQM platform in several different applications, and we present results regarding BGP flaps, PIM flaps in Multicast VPN service, and end-to-end throughput degradation in content delivery network (CDN) service.**

***Index Terms*—Network management, root cause analysis (RCA), service quality management (SQM).**

I. Introduction

AN INCREASINGLY diverse set of applications relies on IP networks. These applications range from entertainment, such as Internet games and streaming videos, to commercial applications, such as e-commerce and online banking, to even some mission-critical applications, such as emergency 911 over VoIP. For many of these applications, best-effort delivery is no longer an acceptable mode of operation. The networking service offered by Internet service providers (ISPs) must maintain ultrahigh reliability and performance.

The change of service quality expectations has also transformed the way that ISPs conduct network and performance management. Network operators have traditionally managed networks on the basis of individual network elements, for example, by detecting and replacing faulty network line cards. Today, managing issues related to end-users' service quality has become an increasingly significant part of operators day-to-day work. This work typically involves such tasks as monitoring the loss and delay among different sites of a customer virtual private network (VPN) and identifying ("alarming"), troubleshooting, and fixing any detected performance problems.

As another example, previously network operators primarily focused on faults and hard failures. Nowadays, their attention is increasingly drawn to transient problems, as is often the case with protocol (e.g., BGP) flaps. By their nature, transient problems "repair" themselves. Therefore, in addition to alarming and responding to each individual problem, examining them—potentially a large number of them—collectively, classifying their root causes, and trending them over time can provide operators with critical insights. This information may help in driving the corresponding failure modes out of the network and may eventually lead to service improvements.

Moreover, as new services (e.g., multicast VPN), new technologies (e.g., MPLS TE), and new devices (e.g., OC768 line cards) are introduced into ISP networks at a fast rate, network operators and hardware vendors often have to learn through experience about service-impacting issues. Should unexpected failure modes or performance impairments occur, operators need to act quickly to understand the problem, diagnose the root cause(s), and eliminate or mitigate the failure mode to improve service quality.

Given these new challenges, traditional fault diagnoses and root cause analysis (RCA) systems [1]–[7] that network operators have relied on are reaching their limits for the following four reasons.

First, the narrow view provided by the per-network element perspective of the traditional systems tends to miss rather complicated *service dependency relationships*. For example, the quality of a VoIP call across the ISP network depends on the status (congestion level, bit error rate, etc.) of the routers and links along the network path carrying the traffic, which is dynamically determined based on the link weights at the time. In another example, the health of a BGP session connecting to a peer router depends on the route processor resource on both routers and the layer-2 line protocol status between them (with complicated timer/protocol interactions), which in turn depends on the condition of the layer-1 (e.g., a SONET ring) network

Manuscript received March 25, 2011; revised October 05, 2011 and January 11, 2012; accepted January 16, 2012; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor R. Teixeira. Date of publication March 16, 2012; date of current version December 13, 2012.

H. Yan, L. Breslau, Z. Ge, D. Pei and J. Yates are with the AT&T Labs—Research, Florham Park, NJ 07932 USA (e-mail: yanhe@research.att.com; breslau@research.att.com; gezihui@research.att.com; peidan@research.att.com; jyates@research.att.com).

D. Massey is with Colorado State University, Fort Collins, CO 80523 USA (e-mail: massey@cs.colostate.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNET.2012.2188837

in between. Capturing such service dependency relationships is vital for service quality management (SQM).

Second, traditional information-gathering processes (such as running *traceroute* or invoking *show* command on routers) that are effective at diagnosing problems for large ongoing service impacting events are unable to cope with minor and transient service disruptions. RCA for transient failures should only rely on proactively collected data.

Third, achieving ultrahigh service quality requires going beyond break-and-fix operation and single-event troubleshooting. SQM involves processing and extracting actionable information from a large number of service impacting events in the aggregate. For example, when analyzing sporadic packet losses observed by probing traffic transmitted between different points of presence (PoPs) of an ISP network, one should examine the packet losses over an extended period (e.g., a month) and diagnose their root causes. Should link congestion be determined to be the primary root cause, capacity augmentation is needed along the corresponding network path. Alternatively, if packet losses are found to be largely due to intradomain routing reconvergence, deploying technologies such as MPLS fast reroute becomes a priority.

Finally, in an ever-changing network and service environment, domain knowledge and operator's experience may become insufficient. RCA systems that solely rely on expert input can fail to capture unexpected service dependencies, which unfortunately are not unusual in practice due to the variety of hardware/software errors and configuration mistakes that can occur. An SQM system should allow rapid instantiation of new RCA tasks based on existing expert knowledge as well as flexible data exploration and data mining capability to improve operators' domain knowledge and understanding over time.

In this paper, we introduce our Generic Root Cause Analysis platform (G-RCA) that is designed to bridge the gap between ISP operational needs for service quality management and the state-of-the-art research [1]–[4] or commercially available [5]–[7] RCA systems. A key feature of G-RCA is a comprehensive service dependency model that includes network topological and cross-layer relationships, protocol interactions, and routing and control plane dependencies. Thus, network operators can look for undesirable network conditions that are potentially *related* to service-impacting problems without specifying the details of the topology and cross-layer relationships, the protocol interactions, or routing dependencies.

We also ensure that the service dependency relationships in G-RCA can be determined using only data that are proactively collected. For example, network paths can be computed from BGP and OSPF route-monitoring data, as opposed to requiring multiple *traceroutes*.

We implemented G-RCA for a tier-1 ISP network. Our design decomposes the RCA process into signature identification for *symptom* and *diagnostic* events, temporal and spatial event correlation, and reasoning and inference logic. Here, symptom events are the service problems to be analyzed, and diagnostic events refer to the evidence of a potential root cause. We define a simple yet flexible rule specification language that allows operators to quickly customize G-RCA into different RCA tools as new problems need to be investigated and understood. We integrate into G-RCA data trending, manual data exploration, and statistical correlation mining capabilities that are tailored for service quality management. G-RCA has proven to be a highly effective SQM platform in several different applications. In particular, using the G-RCA platform, network operators are able to quickly investigate new service problems, uncover unexpected service impacts, and quantify the scale and trend of different factors contributing to service performance issues using the G-RCA platform.

Our contributions can be summarized as follows.

1) We addressed the need for large-scale service quality management in IP networks and services and designed an abstraction model that incorporates complicated service dependency relationships without exposing unnecessary details to network operators.
2) We implemented a G-RCA system for an ISP network by using the data already collected from various logging and performance monitoring systems. We included in our implementation a library of event definitions (for common network problems or failure conditions), network topology and cross-layer conversion utilities, service dependency inference tools, and a library of dependency relationship rules to quickly instantiate new RCA applications.
3) We collaborated with network operators in applying G-RCA in real-world network operations and conducted troubleshooting and analysis for a wide range of problems including customer BGP flaps, cross-site VPN PIM session flaps, and content delivery network (CDN) service performance issues.
4) We demonstrated that iteratively applying RCA and statistical correlation tests is an effective way to identify unexpected network behavior and build RCA rules.

The rest of this paper is organized as follows. In Section II, we first discuss the overall architecture of G-RCA, and we then provide the design details of each component of G-RCA. Section III describes how we quickly incorporate new RCA applications into G-RCA. We demonstrate this by incorporating applications for BGP flaps, throughput drops in CDN service, and PIM flaps in Multicast VPN. Section IV presents the operational experience gained by applying G-RCA in SQM of a tier-1 ISP. Section V discusses related work, and Section VI concludes the paper.

II. G-RCA Architecture

SQM presents a unique set of challenges for ISP networks. This section presents our design of the G-RCA platform for SQM. We focus on the following aspects of G-RCA: 1) data collection and management; 2) the service dependency model; 3) spatial-temporal correlation; 4) reasoning logic; and 5) domain knowledge building. The architecture of G-RCA is shown in Fig. 1.

G-RCA obtains both the symptom events of interest and the set of diagnostic events via the data collector. For a given symptom event, G-RCA uses an application-specific diagnosis graph to identify the relevant diagnostic events. Specifically, G-RCA determines where and when to look for diagnostic events based on the location and time of the symptom event.

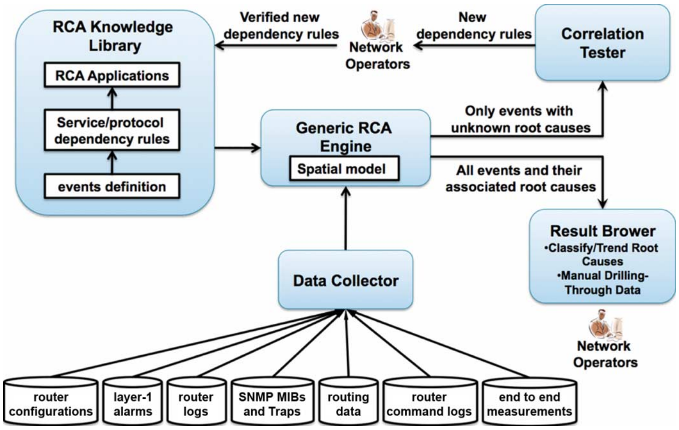


Fig. 1. G-RCA architecture.

Once these diagnostic events are identified, G-RCA then applies reasoning logics to examine all of the different diagnostic events observed for the given symptom to identify the most likely explanation(s) of the symptom event. Operators usually start with an inaccurate and incomplete diagnosis graph and G-RCA allows them to gradually acquire new knowledge or learn unexpected network behaviors to improve the diagnosis graph.

Overall, there are two types of scenarios in which G-RCA is frequently applied.

1) *Troubleshooting individual ongoing network incidences:* These incidences may currently be impacting customers, in which case network operators are under great pressure to quickly go through a large number of alarms, logs, and measurement data and identify the root cause.
2) *Investigating past behaviors in order to improve future network performance:* Besides critical faults and service interruptions, there are many noncritical outages or undesirable conditions in the network. Some are very short in duration, such as a link flap that clears itself before a human operator can get to it. Some are minor in severity, such as a router processor becoming temporarily overloaded, increasing the risk for protocol malfunction, or end-to-end monitoring system reporting sporadic packet losses across the network. These "small" incidences or service impairments can add up, becoming a chronic issue and causing customer dissatisfaction. It is critical for network operators to keep track of a potentially overwhelming number of "small" network events and analyze their root causes so operators can prioritize efforts to improve the network. For example, if link congestion is determined as the primary root cause for packet losses reported from end-to-end monitoring systems, capacity should be added to the corresponding network path. Or if packet losses are found to be largely due to intradomain routing reconvergence, perhaps priority should be put on deploying technologies such as MPLS fast reroute.

A. Data Collection and Management

Understanding service quality issues often requires an integrated view of different parts of the network. As mentioned earlier, G-RCA relies on a wide range of *proactively* collected information containing alarms, logs, and performance measurement data from various network management systems. As simple as it sounds, there are tremendous instrumentation challenges for data management. Moreover, these data come from many devices and network management systems provided by different vendors, all reporting different statistics, from different time zones, and at varying intervals. The same device may be referenced in different ways by different systems or at different network layers (by a circuit identifier, an IP address, or an interface name). The timestamps can be a mixture of local time (depending on the time zone of the device), network time as defined by the service provider, and GMT. To facilitate SQM, one has to look across data sources efficiently. Hence, in G-RCA, the first optimization is on data integration—G-RCA's Data Collector pulls all the data together, normalizes them so that they can be readily correlated, and stores them in database tables in real time. The normalization across naming conventions, time zones, and identifiers takes place as data is ingested into the Data Collector. This hides the data processing complexity from the remaining G-RCA components and eliminates the need for the operators to be painfully aware of the original data source details when correlating data. The data sources in our implementation of G-RCA include layer-1 alarms, router logs, SNMP MIBs and traps, routing data, router command logs, end-to-end measurements, and router configurations. Currently, the Data Collector is collecting around 600 data sources in total, and the daily data volume is about 7 TB. Using two data sources (syslog and SNMP) as examples, the daily numbers of new records for them are tens of millions and hundreds of millions, respectively.

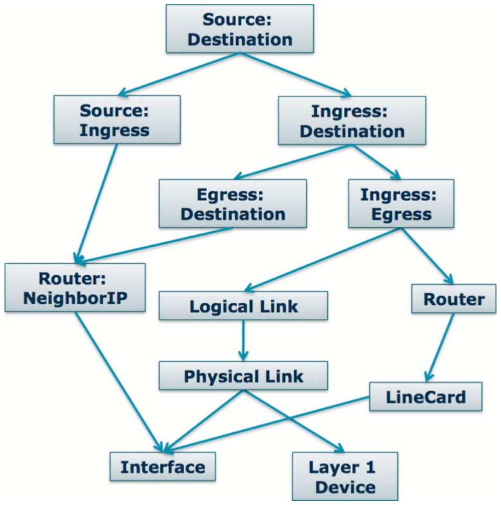


Fig. 2. Spatial model: location types and mapping.

Expectedly, raw data are typically difficult to work with. In G-RCA, we introduce the notion of *event*—an event is a *signature* that captures a particular type of network condition. We associate a *location type* with each event as it provides a key piece of information required for modeling service dependency (in Section II-B). Fig. 2 shows the location types that can be associated with a single event. A type of event can be extracted from raw input data through a parsing script, a database query, or

TABLE I
COMMON EVENT DEFINITIONS FOR A TIER-1 ISP'S IP NETWORK

Event Name	Event Description	Location Type	Data Source
Router reboot	router was rebooted	router	syslog
CPU high (average)	$\geq$ 80% average utilization in 5-minute intervals	router	SNMP
CPU high (spike)	$\geq$ 90% average utilization over the past 5 seconds	router	syslog
Interface down	LINK-3-UPDOWN msg	interface	syslog
Interface up	LINK-3-UPDOWN msg	interface	syslog
Interface flap	LINK-3-UPDOWN msg	interface	syslog
Line protocol down	LINEPROTO-5-UPDOWN msg	interface	syslog
Line protocol up	LINEPROTO-5-UPDOWN msg	interface	syslog
Line protocol flap	LINEPROTO-5-UPDOWN msg	interface	syslog
Regular optical mesh network restoration	regular restoration events in layer-1 optical mesh network	layer-1 device	layer-1 device log
Fast optical mesh network restoration	fast restoration events in layer-1 optical mesh network	layer-1 device	layer-1 device log
SONET restoration	restoration events in the layer-1 SONET network	layer-1 device	layer-1 device log
Link congestion alarm	$\geq$ 80% link utilization in 5-minute intervals	interface	SNMP
Link loss alarm	$\geq$ 100 corrupted packets in 5-minute intervals	interface	SNMP
OSPF re-convergence event	link weight update in OSPF	interface	OSPF monitor
Router Cost In/Out	Router cost in/out inferred from link weight changes	router	OSPF monitor
Link Cost Out/Down	Link cost out or link down inferred from link weight changes	interface	OSPF monitor
Link Cost In/Up	Link cost in or link up inferred from link weight changes	interface	OSPF monitor
Command to Cost In Links	Command typed by operators to cost in links	interface	TACACS
Command to Cost Out Links	Command typed by operators to cost out links	interface	TACACS
BGP egress change	BGP next hop to some external prefix changed	ingress:destination	BGP monitor
In-network delay increase	delay increase between two PoPs	ingress:egress	performance monitor
In-network loss increase	loss increase between two PoPs	ingress:egress	performance monitor
In-network throughput drop	throughput drop between two PoPs	ingress:egress	performance monitor

some more sophisticated processing such as through an anomaly detection program. Specifically, an *event definition* in G-RCA is a tuple consisting of (*event-name, location type, retrieval process, additional descriptive information*), in which the retrieval process points to the actual scripts/queries needed to obtain the matching event instances.

Each *event instance* consists of an (*event-name, event start-time, event end-time, event location, additional info*). For example, the event definition (*link-congestion, interface, myscript*) indicates that the G-RCA Engine will use *myscript* to query SNMP traffic counter data to identify links that are nearly 100% utilized, and output event instances with location type *interface*. A corresponding event instance example is (*link-congestion*, 2010-01-01 12:30:00, 2010-01-01 12:35:00, *newyork-router1:serial-interface0*).

In order for network operators to quickly analyze new service problems, G-RCA defines and implements a wide range of commonly used event signatures. These are included in the RCA Knowledge Library. For example, various RCA applications running on the IP backbone network may be interested in identifying link congestion events. Furthermore, there can be multiple signatures defined for the same network conditions. For example, in G-RCA Knowledge Library, a link congestion event is defined as either a near-100% link utilization in the SNMP traffic counter or a high number of overflow packets in the SNMP interface MIB. The number of overflow packets is a more reliable metric to reflect packet loss as the impact of link utilization on packet loss depends the network type. In the backbone network, as traffic is highly aggregated, there is rarely any packet loss (overflow) even for links with 5-min average utilizations around 90% level. However, it can be expected that in the access network where traffic is more bursty, packet loss can occur with significant lower link utilization measure, hence impacting TCP performance. Network operators can pick the event definition that is best suited for the SQM task under investigation.

Table I lists some common events in G-RCA for the tier-1 ISP network. Note that any event defined in the Knowledge Library can be redefined by an application. For example, the event "link congestion alarm" in Web-hosting data throughput analysis can be easily redefined as "$\geq$ 90% link utilization in the SNMP traffic counter" when needed. At the time of writing this paper, there are more than 200 common events that are defined in the G-RCA Knowledge Library.

B. Service Dependency Model

The key to SQM is understanding the service dependency relationship between a user's service problem and the underlying network devices and protocols supporting the service. G-RCA uses the model in Fig. 2 to capture such dependencies. Though it appears simple, this model actually incorporates topological information (e.g., physical link connecting two different routers), cross-layer dependency (e.g., layer-1 devices supporting layer-3 links), logical and physical device association (which requires router configuration), and dynamic routing (e.g., BGP and OSPF routing in determining the path between source and destination).

The service dependency abstraction is the most powerful component of G-RCA. By specifying the type of service problem (e.g., Ingress:Destination[1]), G-RCA can automatically expand the service dependency to include all network elements that are associated with the service. However, realizing this model in practice is quite challenging. One crucial aspect to the dependency model is that the relationship is time-varying—egress points to a destination network can change upon BGP updates; network paths can change as operators modify link weights; logical to physical mappings can change with configuration changes; even physical connectivity can change over time. Associating the right network elements

[1] In the paper, the notation ”A:B” denotes all locations between points A and B.

with a service event at a given time in history requires reconstructing the "network condition" at the time. G-RCA tackles this by implementing a range of sophisticated conversion utilities as follows.

1) A source and destination pair where both are outside the ISP is first mapped to "Source:Ingress router" and "Ingress router:Destination." This mapping is typically done by looking at the traffic sampling data (e.g., NetFlow) to figure the ingress router and sometimes needs external mapping information. For example, if the traffic enters the ISP's network from a data center that is also under the ISP's control, the mapping can be easily obtained by looking at the configuration (e.g., the list of ingress routers that connect to the data center). Then, in order to map from "Ingress router:Destination" to "Ingress router:Egress router" and "Egress router:Destination," G-RCA looks up historical data of BGP tables to find out the longest prefix match and the network egress point for the destination. Note that BGP routing changes are typically not available at all ingress routers, and only those changes at the BGP route-reflectors are available. In such cases, approximation is needed. The reflectors that feed the ingress router with BGP updates are extracted from the daily archive of router configurations; the BGP decision process at the ingress router is emulated based on the BGP route changes from its reflectors as well as the OSPF distance to available egress routers, and one best egress router is picked based on BGP best path selection.
2) Both "Source:Ingress" and "Egress:Destination" can be mapped to a pair of access router and neighbor's IP according to router configuration. The mapping from "Router: NeighborIP" to "Interface" can also be acquired by looking at the router configuration. This is particularly useful for diagnosing some protocol (e.g., BGP) events with a neighbor IP that typically belongs to a router outside the ISP network.
3) Given the ingress router to egress router pair, the logical link or router level path between them can be computed via an OSPF [8] routing simulation based on network-wide link weights from route-monitoring tools such as OSPFMon [9] (which listens to the flooded messages in OSPF). In the case of Equal Cost Multipath (ECMP), all network elements along all paths will be considered.
4) A point-to-point logical link can be associated with its attached routers by matching the IP addresses of the logical interface to a /30 network.
5) A logical link may be mapped to more than one physical link for redundancy and capacity purposes by using techniques such as SONET Automatic Protection Switching (APS)[10] and Multilink PPP bundle [11]. This mapping can be obtained from the router configuration.
6) G-RCA parses daily router configuration snapshots to infer that a router consists of a set of line-cards, which comprises multiple interfaces.
7) An external database that keeps track of layer-1 inventory provides G-RCA with the mapping from physical links to all the layer-1 devices in between.

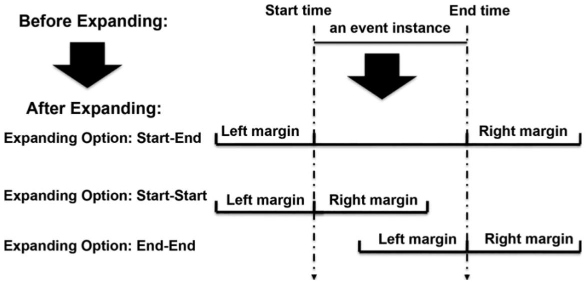


Fig. 3. Three different expanding options.

These conversion utilities are specific to the ISP network that we work with. However, we believe similar capability can be established when applying G-RCA to other networks.

C. Spatial-Temporal Correlation

The most commonly asked question when network operators perform SQM tasks is *what happened in the network at the time that can be related to the service problem?* Breaking this question into more rigid and programmable logic, G-RCA defines a temporal and spatial join rule as follows.

The simple concept "at the same time" can be quite entangled with each networking application. First, there are typically various delay timers or expiration timers in each network protocol. Cause and effect rarely follow one another instantly. Second, there is always inaccuracy and uncertainty in the timing of network measurements. For example, a router CPU measurement in a 5-min interval (via SNMP) may indicate a CPU overload condition within that interval, but not any more precisely. G-RCA captures the above by defining a time window to allow symptom event and diagnostic event to be joined (or "at the same time").

Specifically, each temporal joining rule consists of six parameters: the left expansion margin X, right margin Y, and an expanding option (Start/End, Start/Start, or End/End) for each of the symptom event and diagnostic event. The margin values can be positive or negative in seconds, indicating forward shift or backward shift in time. The expanding option (Fig. 3) specifies how the time window of an event is expanded. G-RCA determines a joint event pair when their expanded time windows overlap. Note that typically operators decide the parameters based on their domain knowledge.

For example, consider a diagnosis event "Interface flap" (Start/End, $\mathrm{X} = 5, \mathrm{Y} = 5$) to be correlated with a symptom event "eBGP flap" (Start/Start, $\mathrm{X} = 180, \mathrm{Y} = 5$). Here, 180 is used to model the cause–effect delay between "eBGP flap" and "Interface flap." The default setting for the eBGP hold timer is 180 s. In other words, "eBGP flap" is likely to occur 180 s after an "Interface flap" event takes place. To model the inaccurate timestamps in syslog messages, 5 s is used. For an "eBGP flap" starting at time 1000 and ending at time 2000, its expanded time interval is $[820, 1005]$. For an "Interface flap" starting at time 900 and ending at time 901, its expanded time interval is $[895, 906]$. The two event instances are considered temporally joined since the two time intervals overlap.

For a diagnostic event to be correlated with a symptom event spatially, G-RCA defines the spatial joining rule that consists

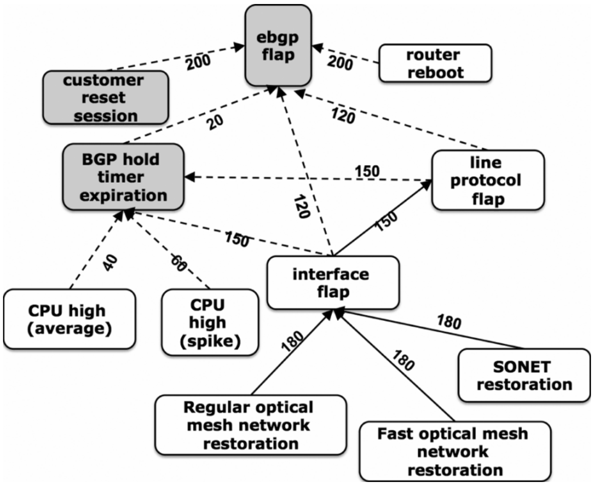


Fig. 4. Diagnosis graph for BGP flaps root cause analysis.

TABLE II
COMMON DIAGNOSIS RULES FOR A TIER-1 ISP'S NETWORK

Symptom Event	Diagnostic Event
Line protocol down/up/flap	Interface down/up/flap
Interface down/up/flap	SONET restoration
Line protocol down/up/flap	SONET restoration
Interface down/up/flap	Regular optical mesh network restoration
Line protocol down/up/flap	Regular optical mesh network restoration
Interface down/up/flap	Fast optical mesh network restoration
Line protocol down/up/flap	Fast optical mesh network restoration
BGP egress change	Interface down/up/flap
BGP egress change	Line protocol down/up/flap
Edge-to-edge delay increase	BGP egress change
Edge-to-edge loss increase	BGP egress change
Edge-to-edge throughput drop	BGP egress change
Edge-to-edge delay increase	Link congestion alarm
Edge-to-edge loss increase	Link congestion alarm
Edge-to-edge throughput drop	Link congestion alarm
Edge-to-edge delay increase	OSPF re-convergence event
Edge-to-edge loss increase	OSPF re-convergence event
Edge-to-edge throughput drop	OSPF re-convergence event
Link loss alarm	Link congestion alarm
Link loss alarm	Line protocol down/up/flap
OSPF re-convergence event	Line protocol down/up/flap
OSPF re-convergence event	Interface down/up/flap
OSPF re-convergence event	Commands to Cost In/Out Links
Link Cost Out/Down	Line protocol down
Link Cost Out/Down	Interface down
Link Cost Out/Down	Command to Cost Out Links
Link Cost In/Up	Line protocol up
Link Cost In/Up	Interface up
Link Cost In/Up	Command to Cost In Links
Link congestion alarm	OSPF re-convergence event

of three parts: 1) symptom event location type; 2) diagnostic event location type; and 3) *joining level*. The first two follow directly from the event definitions and must be one of the location types specified in Fig. 2. The *joining level* is used to link symptom event locations with diagnostic event locations. G-RCA automatically converts the locations of symptom and diagnostic events into the same "join level" location so that they can be directly compared. For example, the symptom is an end-to-end packet loss event that has a location type as "Source: Destination." The diagnostic event is a CPU overload event that has a location type as "Router." The joining level can be "Backbone Router-level Path," which means only CPU overload event on the router along the backbone path (not all the routers on the backbone) will be joining with this end-to-end packet loss event. As another example, consider the symptom event of "packet loss on the uplink of an access router"[2] with location type "Interface." Consider the diagnostic event of "packet loss on an ISP access router customer-facing interface" also with location type "Interface." If the joining level is "Router," two event instances are spatially joined only if they take place on the same router. The Generic RCA Engine evaluates the built-in spatial model that ensures the symptom and diagnostic events are related according to the spatial joining rule specified. With this capability, when building a new application from G-RCA, operators are alleviated from the details of routing information, network topologies, router configurations, and cross-layer dependency.

The above defines the temporal and spatial relationship between a pair of symptom and diagnostic events. For any RCA application, typically many diagnostic signatures are investigated as potential root causes. We model this using a *diagnosis graph*—an example of a diagnosis graph is shown in Fig. 4. We refer to each edge in the diagnosis graph (the pair of symptom and diagnosis events and their temporal and spatial joining rules) as "*diagnosis rule*." Given a diagnosis graph (for a specific SQM application), G-RCA evaluates the time and location conditions and collected data according to the data *retrieval process* in the event definition to determine the presence or the absence of diagnostic signature events.

Similar to the event definition library for frequently used event signatures, G-RCA also includes a library of diagnosis rules in G-RCA Knowledge Library in Fig. 1. Some statistical correlation tests such as [12] can help operators find more diagnosis rules automatically. Note that even with these statistical correlation tests, domain knowledge is still indispensable to check if the identified rules are actually meaningful and to decide the right parameters for the rules. At the time of writing this paper, there are more than 300 common diagnosis rules that are defined in the G-RCA Knowledge Library as shown in Table II.

D. Reasoning Logic

Once data are collected regarding the presence or absence of diagnostic signature events, the next step is to determine the root cause of the symptom events based on this "evidence." This reasoning logic can be implemented in many ways. In particular, G-RCA includes two reasoning engines: rule-based decision-tree-like reasoning and Bayesian inference.

1) Rule-Based Reasoning Module: In our rule-based reasoning module, we allow operators to associate a priority value for each edge in the diagnosis graph (such as in Fig. 4). The higher the priority value, the stronger support that the operator believes the diagnostic event to be the real root cause. After temporal spatial correlation, each symptom event instance is at the root of the diagnosis graph, and diagnostic event instances are located at other nodes of the diagnosis graph. The rule-based

[2] An uplink is the link that connect an access router to a core network router.

reasoning engine starts from the root, searches through each node (if there is a diagnostic event instance), and identifies the leaf node with the maximum priority as the root cause. In the case of a tie between different leaf nodes, all of them are output as joint root causes.

Regarding the priorities of root causes, G-RCA relies on the domain knowledge from operators. In general, the priority assignment for the root causes on the same branch is trivial; operators just need to make sure the deeper root cause has a higher priority. For example, event Interface flap and event line protocol flap can both be the root cause of symptom event BGP flap. Because line protocol flap is typically caused by Interface flap, the priority for Interface flap is higher. It is more tricky to assign priorities for the unrelated root causes on different branches. For example, the priorities for ”Router reboot,” CPU overload, and Interface flap are purely determined by operators according to their knowledge about which one is more likely to the real root cause of BGP flap. If operators are not sure about which root cause is more likely, they simply use the same priority for all of them. G-RCA's Result Browser allows them to exam all potential root causes ordered by the priority.

2) Bayesian Inference: An alternative to the classic priority and rule-based reasoning is the Bayesian inference technique, which has proven successful in many networking applications [13]–[17]. While it is considerably more complex in parameter setting (a drawback based on operators feedback), including Bayesian inference in G-RCA provides several key advantages. For example, it naturally models unobservable root cause conditions (i.e., those that do not have strong observable evidence or signatures) and captures the uncertainty of diagnostic evidence. Using Bayesian inference also allows multiple symptom events to be examined together and deduces a common root cause (or causes) for them—this typically achieves better accuracy than when each individual symptom event is diagnosed separately.

We model the root cause analysis problem using a Naive Bayesian Classifier [18], in which the potential root causes are the *classes*, and the presence or absence of the diagnostic evidence as well as the symptom events themselves are the *features*. The likelihood for a particular root cause r given the features observed $(e_1, \ldots, e_n)$ is

$$p(r|e_1, \ldots, e_n) = \frac{p(r)p(e_1, \ldots, e_n|r)}{\sum\limits_{r \in R} p(r)p(e_1, \ldots, e_n|r)} \tag{1}$$

where R is the set of potential root causes. Determining the root cause is to identify the one producing the following maximum *likelihood ratio*:

$$\arg\max_{r \in R} \frac{p(r|e_1, \ldots, e_n)}{p(\bar{r}|e_1, \ldots, e_n)} = \arg\max_{r \in R} \frac{p(r)}{p(\bar{r})} \times \frac{p(e_1, \ldots, e_n|r)}{p(e_1, \ldots, e_n|\bar{r})} \tag{2}$$

in which $\bar{r}$ denotes when the root cause is not r.

Suppose operators want to assess the likelihood ratio for a BGP session flap due to an overloaded router CPU. In this case, $p(r)$ is the *a priori* probability of an overloaded router CPU inducing BGP session timeout. $p(e_1, \ldots, e_n|r)$ is the probability of the presence of evidence (such as SNMP 5-min average CPU measurement being high, or a BGP hold-timer expiry notification observed in router syslog) under such a scenario; it is divided by $p(e_1, \ldots, e_n|\bar{r})$, which is the chance for the same evidence to appear when the BGP flap is due to other root causes. Hence, the first term in (2) quantifies how likely is the root cause without any additional information, and the second term quantifies how much confidence you gain or lose from observing or not observing the set of evidence. When the features are conditionally independent, the second term can be decoupled to $\Pi_i \frac{p(e_i|r)}{p(e_i|\bar{r})}$, in which each term quantifies the support of root cause r given evidence e_i.

The parameters $\left(\text{ratios: } \frac{p(r)}{p(\bar{r})} \text{ and } \frac{p(e_i|r)}{p(e_i|\bar{r})}\right)$ can be difficult to configure. These can be trained from classified historical data, which we can bootstrap using the rule-based reasoning from Section II-C. Alternatively, we also define a fuzzy type of discrete values. Operators can simply specify the ratios as “Low,” “Medium,” and “High,” which corresponds to values 2, 100, and 20 000, respectively. Note that the unscaled values are likely to be fractional numbers less than one as the root causes are rare events. However, from the operational point of view, it is undesirable to use fractional numbers. According to (2), multiplying a constant scaling factor does not change the final results. Thus, instead we use integers like 2, 100, and 20 000. Coarse as they are, the classification results using these are quite reasonable in that the performance of the Naive Bayesian classifier is often not sensitive to the probability parameters [19].

3) Comparison: Interestingly, in our operational practice, we have found that rule-based reasoning logic is often preferred over its more sophisticated counterpart—this is because: 1) it is easier to configure; 2) it gives simple and direct association between the diagnosed root cause and the evidence(s) for result interpretation; and (3) it is found to be very effective in most applications that we have explored. However, there are a few cases where Bayesian inference is preferred—for example, when the root cause condition is unobservable (e.g., no direct evidence can be collected).

E. Domain Knowledge Building

One of the important challenges in SQM is that operators' domain knowledge and operational experience can be unreliable or incomplete. This implies that the specification of a diagnosis graph for a new SQM application offered by an operator, especially the initial version, can be inaccurate and incomplete.

G-RCA assures the accuracy of diagnosis graph by using the Correlation Tester (see Fig. 1) to check each edge/rule in the graph. Specifically, for each diagnosis rule, we run the Correlation Tester to test the statistical correlation between symptom event and diagnostic event. Regarding the Correlation Tester, G-RCA implements the statistical correlation algorithm proposed in NICE [12]. In comparison to other canonical statistical tests, NICE handles the event autocorrelation structure very well, which is commonly observed in networking event series. The diagnosis rule is only considered to be accurate when it passes the test. The idea is that the number of coincidental correlations should be bounded when examining the instances of symptom and diagnostic events in bulk. Thus, when the diagnosis rule is inaccurate, it fails the test due to lack of statistical correlation between symptom event and diagnostic

TABLE III
APPLICATION-SPECIFIC EVENTS FOR BGP FLAPS ROOT CAUSE ANALYSIS

Event Name	Event Description	Data Source
eBGP flap	eBGP session goes down and comes up, BGP-5-ADJCHANGE msg.	syslog
Customer reset session	eBGP session is reset by the customer, BGP-5-NOTIFICATION msg.	syslog
eBGP HTE	eBGP hold timer expired, BGP-5-NOTIFICATION msg.	syslog

event. We also periodically retest each diagnosis rule in the diagnosis graph to keep the diagnosis rules up to date.

G-RCA addresses this concern regarding incomplete diagnosis graph through iteratively using the Correlation Tester and Result Browser (see Fig. 1). G-RCA first allows operators to filter out the symptom events with known root causes with the root cause classification capability provided in the Result Browser. Second, operators are able to focus on the rest of symptom events by comparing with other suspected diagnostic events (regardless if they are not currently defined in the diagnosis graph) that occur at about the same time and that are spatially related to the service problem under investigation. On one hand, the second step can be done via manual drill-down and data exploration capability in the Result Browser. The manual-discovered diagnosis rules need to be tested by the Correlation Test before incorporating into the diagnosis graph. On the other hand, operators can also choose to run the Correlation Tester *blindly* between the symptom events without known root causes and each type of suspected diagnostic events. Note that a relation between the symptom events and one type of suspected diagnostic events might be buried in the noise if we do not take out the symptom event with known root causes. As G-RCA emphasizes usability, the newly uncovered diagnosis rules need to be verified by operators before incorporating into the diagnosis graph. For example, a large number of BGP flaps between customer routers and provider edge routers were found to be due to link flaps between the customer and provider edge routers, which typically are caused by customer activities. These BGP flaps could be easily filtered out by the Result Browser, and operators can concentrate on the rest of the BGP flaps without known root causes. This has proven tremendously useful from operational experience. Operators can often spot the signature of overlooked root causes and add them to the diagnosis graph. Similarly, instead of focusing on the symptom events with unknown root cause, one can concentrate on the symptom events with a particular type of root cause to dig out deeper root causes. For example, by only looking at the BGP flaps caused by ”CPU overload,” one may find the deeper root cause that results in ”CPU overload.”

Through iteratively using the Result Browser and Statistical Correlation Tester, operators can start with inaccurate and incomplete domain knowledge and gradually acquire new knowledge or learn unexpected network behaviors exhibited in the network data, which can then be incorporated into the diagnosis graph.

III. G-RCA APPLICATIONS

The key advantage of G-RCA in SQM is its capability to be rapidly customized into different RCA applications in the ISP's network. Exisitng RCA applications include various diagnostic systems for different types of protocol (e.g., BGP and PIM) flaps, for network issues detected by periodical probing traffic sent across the backbone network, and for degrading service performance conditions in CDN, DNS, and 3G Cellular network. In this section, we focus on the following three case studies—1) customer BGP flaps; 2) end-to-end throughput management in a CDN service; and 3) network PIM flaps in multicast VPN—to demonstrate the effectiveness of G-RCA.

A. BGP Flaps Root Cause Analysis

In the first case study, we focus on building an RCA tool to understand the root causes of eBGP [20] session flaps between customer routers (CRs) and provider edge routers (PERs) in a tier-1 ISP.

Customer networks exchange routes with the ISP through the eBGP session— the routes learned from the ISP inform the customer network how to route to locations across the Internet and other sites of the same customer; routes shared from the customer network ensure that other sites can reach the sites. If a session flaps, all routes are withdrawn, and traffic is disrupted. Although relatively short (on the order of a minute), these flaps can disrupt applications. For example, VoIP sessions may be lost, and financial transactions may be interrupted.

We therefore aim to minimize the number of eBGP session flaps, taking actions to drive avoidable flaps permanently out of the network. The first step to achieving this is to understand the root cause of the flaps—a particularly challenging problem across a trust domain (between customer and provider networks). We achieve this using G-RCA by constructing application specific events and rules.

1) Application-Specific Configuration: We start by constructing our BGP flap-specific events—those events that are not already included in the common event definitions in the RCA Knowledge Library (Table I). These new application-specific events are illustrated in Table III. Note that there are only three of them, in contrast with seven other events that we reuse from the Knowledge Library.

After defining the application-specific events, we need to add a few application-specific diagnosis rules. The complete diagnosis graph is depicted in Fig. 4. This combines events and rules taken from the RCA Knowledge Library (Table II) with BGP application-specific events (shown as gray boxes) and application specific rules (dashed lines). Let us now examine the diagnosis graph (Fig. 4) from bottom to top. Layer-1 events such as fast restoration in optical mesh network and SONET restoration may cause interface flaps on PERs. Furthermore, interface flaps may induce BGP hold timer expirations, line protocol flaps, or even eBGP flaps. If BGP fast external fallover [21] is enabled, an interface flap can directly trigger an eBGP flap without requiring the BGP hold timer to expire. All the CPU utilization related events such as “CPU high (average)” and “CPU high

TABLE IV
ROOT CAUSE BREAKDOWN OF BGP FLAPS

Root Cause	Percentage (%)
Router reboot	0.33
Customer reset session	1.84
CPU high (average)	0.02
CPU high (spike)	6.44
Interface flap	63.94
Line protocol flap	11.15
eBGP HTE (due to unknown reasons)	4.86
Regular optical mesh network restoration	0.04
Fast optical mesh network restoration	0.14
SONET restoration	0.29
Unknown	10.95

(spike)" can only cause eBGP flaps through BGP hold timer expiration. On the top of this figure, we can see the events "router reboot" and "customer reset session" could also cause eBGP flaps.

Finally, we specify priorities for different diagnosis rules for BGP flaps RCA, as depicted via the numbers on the edges in Fig. 4. The highest priority is used to determine the most likely root cause among multiple root causes by the G-RCA Engine. For example, if a BGP flap joins with both a high CPU event and a layer-1 flap, the layer-1 flap is identified as the root cause of this BGP flap as it is associated with a higher priority (180) edge.

2) Results: In order to demonstrate how effectively G-RCA can identify the root causes of BGP flaps in the ISP, we ran the BGP flap RCA tool configured above for more than 600 provider edge routers in different locations, each of which has several hundred eBGP sessions established with customer routers. Table IV shows the root cause breakdown generated by the Result Browser in G-RCA for all the BGP flaps on these provider edge routers during a month.

This RCA application is now an integral part of the BGP monitoring in the tier-1 ISP. It is used to trend flaps and identify anomalous behavior that requires investigation (e.g., behavioral changes after new software upgrades). It is also used by operations and customer service representatives to provide automatic analysis of specific customer BGP flaps for rapid responses to customer inquiries about such events. In the BGP RCA application, the average diagnosis time per symptom event is less than 5 s.

B. Root Cause Analysis for CDN Service Impairments

In this case study, we discuss how to build a new RCA application for troubleshooting service impairments in the ISP's CDN [22]–[25]. The ISP operates a CDN service in which static Web objects are hosted at several data centers across its network. Through dynamic DNS binding, HTTP requests are directed to the "closest" data centers and served from there.

To manage the performance of the CDN service, the traffic monitor passively observes the end-to-end round-trip time (RTT) between end-users and CDN servers as Web service requests arrive.

The primary challenge is to identify the network and service elements involved in servicing the requests at the precise time of the performance degradation. This is challenging to achieve during a real-time event and practically impossible to manually identify for historical events. However, G-RCA's spatial model and proactive data collection enables such determination and is the key to providing the ability to automatically troubleshoot these service issues.

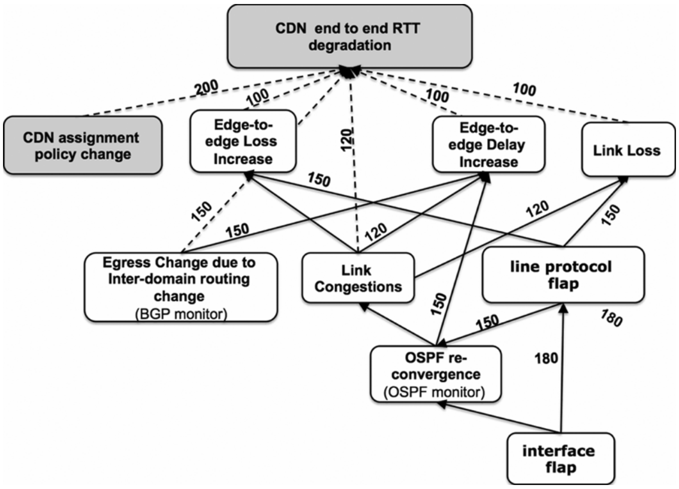

CDN end to end RTT degradation
200
100
120
100
100
CDN assignment policy change
Edge-to-edge Loss Increase
Edge-to-edge Delay Increase
Link Loss
150
150
150
120
120
150
Egress Change due to Inter-domain routing change (BGP monitor)
Link Congestions
line protocol flap
150
150
180
OSPF re-convergence (OSPF monitor)
180
interface flap

Fig. 5. Diagnosis graph for CDN RTT degradation root cause analysis.

1) Application-Specific Configuration: To create the RCA application for CDN service impairments, we defined the application-specific events (Table V) and diagnosis rules (dashed lines in Fig. 5). Note that the majority of the events and rules could again be drawn from the RCA Knowledge Library. The most important application-specific event is the "CDN end-to-end throughput drop" inferred from Keynote measurements. This event indicates a decrease in average download throughput and is the input to the RCA application. Each "CDN end-to-end throughput drop" event is associated with a start time and a location, which is defined by the CDN server and client machine (e.g., Keynote agent) pair. In addition to analyzing the performance event generated from Keynote measurements, the RCA application also allows operators to directly enter an event of interest conforming to the above format. This greatly improves the flexibility of the tool as there exist many channels to detect a service performance problem other than Keynote, such as through a customer service call.

After defining the application-specific events, we then need to add a few application-specific diagnosis rules, which are not already included in the RCA Knowledge Library (Table II). As shown in Fig. 5, a few application-specific diagnosis rules and some other rules from the RCA Knowledge Library together form a full diagnosis graph for root cause analysis of service impairments in the CDN. Note that in Fig. 5, application specific events are shown in gray boxes and application-specific rules are shown with dashed lines. As with the BGP flaps case study, priorities (numbers on the edges) for different diagnosis rules are also defined in Fig. 5.

2) Results: We evaluated this RCA application with all the RTT degradation events in one month observed between millions of users and a particular northeast CDN node. Table VI shows the root cause breakdown generated by the Result Browser in G-RCA. At a high level, only 25.17% of the RTT

TABLE V
APPLICATION-SPECIFIC EVENTS FOR ROOT CAUSE ANALYSIS OF ROUND-TRIP TIME INCREASE IN CDN

Event Name	Event Description	Data Source
CDN round trip time increase	increase in end-to-end round trip time (RTT) between end-users and CDN servers	
CDN server issue	CDN server load is high	server logs

TABLE VI
ROOT CAUSE BREAKDOWN OF END-TO-END RTT DEGRADATIONS

Root Cause	Percentage (%)
CDN assignment policy change	3.83
Egress Change due to Inter-domain routing change	5.71
Link Congestions	3.50
Link Loss	3.32
Interface flap	4.65
OSPF re-convergence	4.16
Outside of our network (Unknown)	74.83

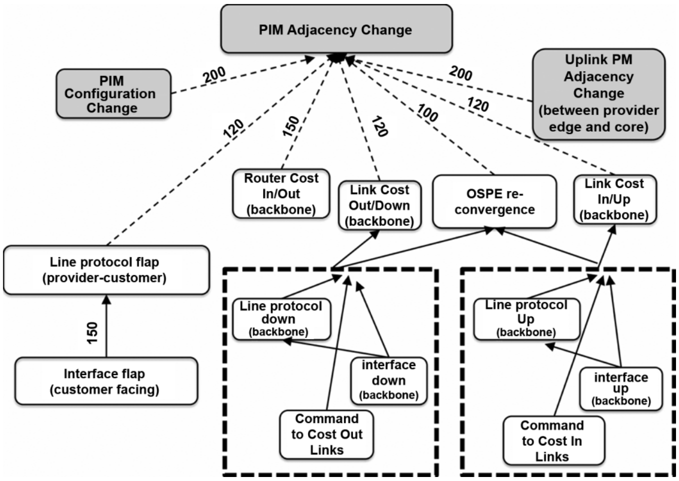

PIM Adjacency Change
PIM Configuration Change
200
Uplink PM Adjacency Change (between provider edge and core)
200
120
100
120
150
120
Router Cost In/Out (backbone)
Link Cost Out/Down (backbone)
OSPE re-convergence
Link Cost In/Up (backbone)
Line protocol flap (provider-customer)
150
Interface flap (customer facing)
Line protocol down (backbone)
interface down (backbone)
Command to Cost Out Links
Line protocol Up (backbone)
interface up (backbone)
Command to Cost In Links

Fig. 6. Diagnosis graph for PIM adjacency change root cause analysis.

degradations are identified as caused by either events that happened within our network (e.g., interface flap, link congestion, and CDN assignment policy change) or from events that are visible in our network (such as BGP routing changes announced by other ISPs). For the rest (majority) of them, we did not find any evidence from inside our network, which suggests that those RTT degradations may be caused by other ISPs on the end-to-end path.

According to the CDN service operations team, this RCA application is quite useful in finding the root causes and is much more rapid than could be achieved by Operations personnel. As an example event, the RCA application successfully determined that a given RTT degradation was caused by the failure of the peering link between our network and the neighboring ISP. This failure resulted in a routing change, which in turn resulted in traffic experiencing larger delays and degraded TCP performance. Although the network operations team was already aware of the peering link failure and working on it, the CDN service operations team could in parallel repair service even before the network was repaired by updating the DNS tables to route impacted users to “closer” CDN nodes as measured by the new network routing. Thus, the two teams could work in parallel—with CDN performance being repaired even while the network issue was still being resolved. Having rapid root cause analysis through G-RCA thus enables faster intervention on customer-impacting issues and fast service improvement.

In the CDN RCA application, the average diagnosis time per symptom event is less than 3 min. Most of the delay is incurred computing interdomain (BGP) routes and intradomain (OSPF) routes.

C. Root Cause Analysis of PIM Adjacency Change in Multicast VPN

In the final case study, we describe the use of G-RCA to identify the root cause of problems within a Tier-1 ISP’s Multicast VPN (MVPN) service. For each MVPN customer, all PERs at which the customer attaches to the provider network maintain Protocol-Independent Multicast (PIM) Neighbor adjacencies with each other using a Hello protocol. The loss of PIM neighbor adjacencies, which is reported via syslog, is often a good indicator of service-related problems. However, not all such changes are indicative of an actual problem (e.g., some are due to customer disconnects). Due to the sheer volume of these messages (thousands per day), manual analysis to determine the root cause of each event to determine those which are indicative of an actual problem is infeasible.

1) Application-Specific Configuration: A G-RCA application to determine the root cause of PIM neighbor adjacency changes within the MVPN service was created. The resulting diagnosis graph is shown in Fig. 6. The kinds of root cause events that were determined to have caused PIM neighbor adjacency changes include router configuration changes, problems on the provider–customer link, routing changes within the ISP backbone, and problems on the PER uplinks to the backbone network. Since the application was able to reuse many of the events and rules in the RCA Knowledge Library, we only needed to add three multicast-specific events (Table VII) and seven multicast-specific diagnosis rules (dashed lines in Fig. 6). For example, we reused many events and rules regarding the path changes between a pair of PERs and the unstablities on the provider–customer link. Actual development time was no more than 10 h. Without G-RCA, building a root cause analysis tool for this problem would have required months of development and may not have happened in practice.

2) Results: The PIM RCA application has proved to be extremely useful in classifying root causes of PIM adjacency losses and in guiding operators and engineers to a better understanding of actual MVPN performance in the network, allowing them to focus their effort on those issues that require their attention. Running the G-RCA PIM application on a day’s worth of events required about 1–2 h. For each day, the application is currently able to identify the root causes for more than 98% of PIM neighbor adjacency changes. We expect that with additional attention to those remaining unclassified events, the G-RCA PIM application will determine root causes for more than 99% of the events.

TABLE VII
APPLICATION-SPECIFIC EVENTS FOR ROOT CAUSE ANALYSIS OF PIM ADJACENCY CHANGE IN MULTICAST VPN

Event Name	Event Description	Data Source
PIM Neighbor Adjacency Change	a PE lost a neighbor adjacency with another PE in the MVPN.	syslog
PIM Configuration change	a MVPN is either provisioned or de-provisioned on a router.	router command logs
Uplink PIM adjacency change	a PE lost a neighbor adjacency with its directly connected router on its uplink to the backbone.	syslog

TABLE VIII
ROOT CAUSE BREAKDOWN OF PIM ADJACENCY LOSSES

Root Cause	Percentage (%)
PIM Configuration Change (to add and remove customers)	4.04
Router Cost In/Out	10.34
Link Cost Out/Down	1.50
Link Cost In/Up	0.84
OSPF re-convergence	10.36
Uplink PIM adjacency loss	1.95
interface (customer facing) flap	69.21
Unknown	1.76

In order to demonstrate how effectively G-RCA can identify the root causes of PIM adjacency losses in the ISP, we ran the PIM RCA application configured above for all the PIM neighbor adjacency changes observed in 2 weeks on more than 600 provider edge routers. Table VIII shows the root cause breakdown generated by the Result Browser in G-RCA.

In the PIM RCA application, the average diagnosis time per symptom event is similar to the BGP RCA application, which is typically less than 5 s.

IV. OPERATIONAL EXPERIENCE ON IMPROVING DOMAIN KNOWLEDGE

The main challenge in creating G-RCA applications is identifying the diagnosis rules. Domain knowledge typically provides a solid starting point, but our experience indicated that collating domain knowledge across potentially many domain experts can be surprisingly challenging. Domain knowledge is often distributed across multiple experts—no one expert understands the entire domain. These experts often have trouble thinking of the relevant rules when "put on the spot," or they are so busy fighting issues in the network that it is difficult to obtain their attention for long enough to obtain the information. In other cases, the network operator's domain knowledge may be wrong either because the relationships between events are extremely complex and not well understood, or because the network is not behaving as designed (as in Section III-A.2). We thus found it extremely critical to provide mechanisms integrated in G-RCA to facilitate diagnosis rule learning.

A. Learning Diagnosis Rules via Manual Iterative Analysis

With G-RCA, the individual responsible for creating an RCA application can follow an iterative process to identify new diagnosis rules. For example, in the PIM case, domain experts use data exploratory tools [26] to manually inspect unexplained neighbor adjacency changes and determine root cause(s). Once a new root cause was identified, it was codified in the RCA application, which was then run to identify all those events that could be explained by the augmented set of rules and, more importantly, those that were still unexplained. The domain experts would then further sample remaining unexplained PIM flaps searching for new signatures that could be incorporated. The PIM application developer thus continually whittled down the number of unexplained flaps by iteratively incorporating new rules and examining those that fell outside the scope of the new rules. By using G-RCA's Result Browser, which made individual event analysis easy, the PIM application developer rapidly identified new diagnosis rules for the application and therefore revealed the anomalous behaviors discussed in Section III-C.2.

B. Learning Diagnosis Rules via Statistical Correlation Test

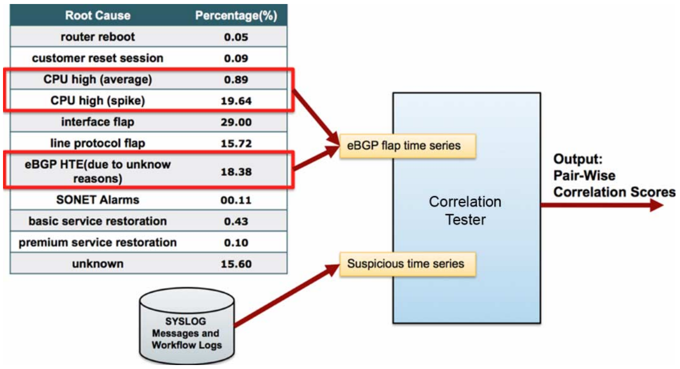
Although the manual iterative analysis was effective in the PIM application, we used a more "intelligent" approach to analyzing BGP flaps. We illustrate this here by discussing our experiences in analyzing BGP flaps that were related to high CPU events.

Table IV illustrated that a significant portion of BGP flaps occurred at the same time as CPU overload was observed on the router. A naive assumption may be that these BGP flaps were in fact induced by high router CPU load. However, further inspection cast doubt on this assumption.

With the integrated data drilling-through functionality implemented in the Result Browser of G-RCA, it is easy for operators to explore additional information such as syslog messages and workflow logs that appear on the same router or location as the event being analyzed. Equipped with the powerful GUI, operators revealed via manual drilling-down that not all BGP flaps with a high CPU signature are actually due to CPU overload on PERs. In most cases, the high CPU utilization is likely *caused by* BGP flaps that are triggered from the customer side. Specifically, a large amount of routing computation on PER in response to the BGP flaps produces high CPU utilization.

With this cyclic causal relationship—"BGP flap causes CPU overload" and "CPU overload causes BGP session timeout"—evidence-based diagnosis systems including our RCA tool hit their limit. We needed further refined signatures such as searching for other potential causes of the high CPU events to identify those that were not BGP-flap-induced and could thus explain BGP flaps.

Rapid manual inspection of events through G-RCA's Result Browser worked well in some situations, but our experience demonstrated that it does not work effectively if looking for relatively rare explanations among a sea of events. Instead, we took a different approach (Fig. 7), using G-RCA's correlation tester module to examine the statistical correlation between CPU-related BGP flaps and other types of events on the same PER. Specifically, we created a time series from all CPU-related BGP flaps as defined by our G-RCA application—those BGP flaps associated with BGP hold timer expiries, but where there was no evidence of link failures that could explain the flap, and which joined with one of the high CPU signatures. We then executed a statistical correlation test [12] between this time series and



Root Cause	Percentage(%)
router reboot	0.05
customer reset session	0.09
CPU high (average)	0.89
CPU high (spike)	19.64
interface flap	29.00
line protocol flap	15.72
eBGP HTE(due to unknow reasons)	18.38
SONET Alarms	00.11
basic service restoration	0.43
premium service restoration	0.10
unknown	15.60

eBGP flap time series
Suspicious time series
Correlation Tester
Output: Pair-Wise Correlation Scores
SYSLOG Messages and Workflow Logs

Fig. 7. Interaction between generic RCA engine and correlation tester.

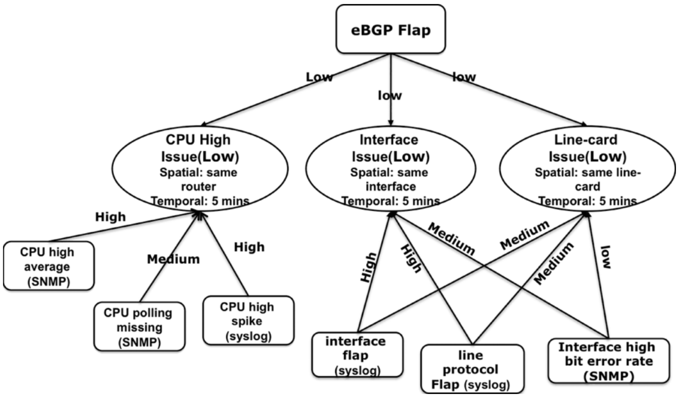
831 other time series created from workflow logs, and 2533 time series from syslog messages.

We fed three months worth of data into the correlation tester to analyze the CPU-related BGP flap. Of the 3361 time series, 80 time series exhibited significant statistical correlation with our CPU-related BGP flaps. A rapid examination of these events by domain experts revealed that many of them were readily explained and/or incorporated into our existing application rules. For example, these CPU-related BGP events were strongly correlated with BGP notifications—a generic message logged for any BGP flap. However, the statistical correlation test did reveal some unexpected correlations. For example, the result revealed that certain provisioning activities (as derived from workflow logs) are strongly correlated with CPU-related BGP flaps. Drilling into individual cases, we identified a small number of incidents where unrelated provisioning activities on some routers appear to have caused customer BGP sessions to flap, an unexpected router software behavior. As a result, 10 such incidents were sent to the router vendor for further investigation; the vendor has since implemented software changes to eliminate this issue.

It is worth noting that the prefiltering of BGP flaps by their root causes as diagnosed by the Generic RCA Engine made a significant difference here. When we fed all BGP flaps to the correlation tester module, the correlation with provisioning activity was no longer statistically significant. By instead focusing on a small subset of the BGP flaps, the correlation "signal" is amplified, revealing the hidden issue. Thus, the interaction between G-RCA engine and the Correlation Tester is crucial to revealing subtle issues.

C. Learning Unobservable Root Causes via Bayesian Inference Engine

Thus far, all examples have been based on rule-based reasoning. We now demonstrate the power of Bayesian inference engine in the G-RCA. In particular, we show how the inference engine can identify a line-card problem as the root cause, which is not readily detectable using rule-based reasoning: A line-card problem is an unobservable root cause as no logs or alarms for line-card issues were incorporated into the RCA tool at the time of our analysis. The configuration for inference engine is shown in Fig. 8. Three virtual root cause events are defined as "CPU High Issue," "Interface Issue," and "Line-card Issue."



eBGP Flap
Low
low
low
CPU High Issue(Low) Spatial: same router Temporal: 5 mins
Interface Issue(Low) Spatial: same interface Temporal: 5 mins
Line-card Issue(Low) Spatial: same line-card Temporal: 5 mins
High
Medium
High
High
High
Medium
Medium
Medium
low
CPU high average (SNMP)
CPU polling missing (SNMP)
CPU high spike (syslog)
interface flap (syslog)
line protocol Flap (syslog)
Interface high bit error rate (SNMP)

Fig. 8. Bayesian inference configuration for BGP flaps RCA application.

We ran both the rule-based reasoning engine and the Bayesian inference engine using one month of eBGP flaps on a PER that has several hundred eBGP sessions. While most of the results are consistent with each other, there are 133 eBGP flaps that the rule-based reasoning engine diagnoses as "Interface flap"-induced. However, the Bayesian inference engine identifies these same flaps as "Line-card Issue"-caused. By manually drilling down into these 133 eBGP flaps (on 125 different eBGP sessions) using G-RCA's Result Browser, we find that all of them are associated with the same line-card and are within 3 min. This is a strong indication for a line-card-related problem. This was later confirmed by network operators, who actually pointed us to a line-card crash signature that was not incorporated into G-RCA's Knowledge Library at the time, and we confirmed that the linecard in question indeed crashed.

Since Bayesian inference easily allows analysis across multiple symptom event instances, the common root cause of these 133 eBGP flaps was successfully inferred.

V. Related Work

Many existing network management systems such as [1]–[7] work on the basis of individual network elements such as routers, line-cards, and interfaces. In contrast, G-RCA focuses on issues related to end-users' service quality such as throughput degradation among different sites of a customer VPN. In addition, most of the existing network management systems focus on faults and hard failures that require immediate investigation, while G-RCA has its primary focus on classifying and trending the root causes of a large number of historical transient events. This provides operators with critical information that would help in driving the corresponding failure mode out of the network and eventually lead to service improvements.

A large body of recent work has focused on root cause analysis of network-layer faults without direct evidence from the lower layer in large ISPs such as SCORE [27], Shrink [16], and [28]. Shared Risk Link Group (SRLG) was proposed to model the cross-layer dependency, where a group of network layers entities depends on the same physical-layer entity. With the concept of SRLG, finding the root cause of network-layer faults becomes a minimal set cover problem in a bipartite graph in SCORE [27] and [28]. Shrink enhanced them by incorporating Bayesian network to model inaccurate measurements and SRLG information. While G-RCA is designed for more general root cause analysis problems, G-RCA could actually incorporate SCORE-like algorithms to infer what is happening if there is no direct evidence.

Machine learning and statistical methods have been widely applied in mining relationships among events. NICE [12] proposed a novel statistical correlation approach with circular permutation test for learning correlation between two event time series. While CORDS [29] employs chi-squared analysis to mine correlations, SPIRIT [30] uses Principal Component Analysis. More sophisticated and computationally expensive techniques such as Hidden Markov Chain [31] and association rule mining [32], [33] have also been proposed to mine relationships among multiple-event time series. Although G-RCA focuses on identifying the root cause of each individual event of interest, these techniques are actually complementary to G-RCA for mining more rules.

VI. Conclusion and Future Work

In this paper, we described G-RCA, a generic root cause analysis platform for service quality management in large IP networks. G-RCA is an ideal platform for SQM in a "constantly changing" network environment. First, it captures the layered network model in its knowledge library in the form of diagnosis rules. These rules can be reused by various RCA applications. Its generic RCA engine implements the common logic found in various RCA tasks such as temporal/spatial correlation, rule-based reasoning, and Bayesian inference. In addition, the generic RCA engine also implements a network location model, which models various network locations and the mappings among them. Thanks to the Knowledge Library and RCA engine, new RCA applications can be quickly incorporated into G-RCA via simple configuration. Second, domain knowledge in existing RCA applications can be refined by the interaction between the RCA engine and the Correlation Tester, which is important for a dynamic network environment. Third, in order to analyze a large number of service quality issues and classify/trend their root causes, it proactively collects all types of data from different sources and normalize them in real time.

Our work can be extended in several directions. First, we plan to make the temporal joining rules less sensitive for robust root cause analysis and deal with the cyclic causal relationship in diagnosis rules. Second, we plan to refine the inference algorithm and simplify its configuration to further improve its usability. Third, we want to support real-time root cause applications. Finally, we will work with network operators to extend the G-RCA platform into other networks and services such as cellular data network, IPTV, and VoIP.

References

[1] P. Corn, R. Dube, A. McMichael, and J. Tsay, "An autonomous distributed expert system for switched network maintenance," *Proc. IEEE GLOBECOM*, pp. 1530–1537, 1988.

[2] C. Joseph, J. Kindrick, K. Muralidhar, and T. Toth-Fejel, "Map fault management expert system," in *Integrated Network Management I*. Amsterdam, The Netherlands: North Holland, 1989, pp. 627–636.

[3] J. Wright, J. Zielinski, and E. Horton, "Expert systems development: The ACE system," in *Expert Systems Applications to Telecommunications*. New York: Wiley, 1988, pp. 45–72.

[4] T. Yamahira, Y. Kiriha, and S. Sakata, "Unified fault management scheme for network troubleshooting expert system," in *Integrated Network Management I*. Amsterdam, The Netherlands: Elsevier, 1989.

[5] Hewlett-Packard, "HP Operations Center," 2011 [Online]. Available: http://www.hp.com/go/ngoss

[6] IBM, "IBM Tivoli," 2011 [Online]. Available: https://www-01.ibm.com/software/tivoli/

[7] EMC Corporation, "EMC Ionix platform," 2011 [Online]. Available: http://www.emc.com/products/family/ionix-family.htm

[8] J. Moy, "RFC2328: OSPF version 2," 1998.

[9] A. Shaikh and A. Greenberg, "OSPF monitoring: Architecture, design, and deployment experience," in *Proc. USENIX/ACM NSDI*, 2004, p. 5.

[10] Cisco, "SONET automatic protection switching," 2011 [Online]. Available: http://www.cisco.com/en/US/tech/tk482/tk606/tsd_technology_support_sub-protocol_home.html

[11] Juniper Networks, "Overview of multilink PPP bundle," 2010 [Online]. Available: http://www.juniper.net/techpubs/software/erx/junose81/swconfig-link/html/mlppp-config2.html

[12] A. Mahimkar, J. Yates, Y. Zhang, A. Shaikh, J. Wang, Z. Ge, and C. Ee, "Troubleshooting chronic conditions in large IP networks," in *Proc. ACM CoNEXT*, 2008, Article no. 2.

[13] I. Cohen, M. Goldszmidt, T. Kelly, and J. Symons, "Correlating instrumentation data to system states: A building block for automated diagnosis and control," in *Proc. OSDI*, 2004, pp. 231–244.

[14] S. Zhang, I. Cohen, M. Goldszmidt, J. Symons, and A. Fox, "Ensembles of models for automated diagnosis of system performance problems," in *Proc. IEEE DSN*, 2005, pp. 31–109.

[15] I. Cohen, S. Zhang, M. Goldszmidt, J. Symons, T. Kelly, and A. Fox, "Capturing, indexing, clustering, and retrieving system history," in *Proc. 20th ACM Symp. Oper. Syst. Principles*, 2005, pp. 105–118.

[16] S. Kandula, D. Katabi, and J. Vasseur, "Shrink: A tool for failure diagnosis in IP networks," in *Proc. ACM SIGCOMM Workshop Mining Netw. Data*, 2005, pp. 173–178.

[17] P. Bahl, R. Chandra, A. Greenberg, S. Kandula, D. A. Maltz, and M. Zhang, "Towards highly reliable enterprise network services via inference of multi-level dependencies," in *Proc. ACM SIGCOMM*, 2007, pp. 13–24.

[18] "Overview of naive Bayes classifier," 2011 [Online]. Available: http://en.wikipedia.org/wiki/Naive_Bayes_classifier

[19] I. Rish, "An empirical study of the naive Bayes classifier," in *Proc. IJCAI Workshop Empirical Methods Artif. Intell.*, 2001, pp. 41–46.

[20] Y. Rekhter, T. Li, and S. Hares, "A Border Gateway Protocol 4 (BGP-4)," 2006 [Online]. Available: http://www.ietf.org/rfc/rfc4271.txt

[21] Cisco, "Cisco IOS BGP command reference," 2011 [Online]. Available: http://www.cisco.com/en/US/docs/ios/iproute_bgp/command/reference/irg_book.html

[22] M. Pathan, R. Buyya, and A. Vakali, "Content delivery networks: State of the art, insights, and imperatives," in *Content Delivery Networks*. New York: Springer, 2008, p. 1.

[23] J. Dilley, B. Maggs, J. Parikh, H. Prokop, R. Sitaraman, and B. Weihl, "Globally distributed content delivery," *IEEE Internet Comput.*, vol. 6, no. 5, pp. 50–58, 2002.

[24] S. Saroiu, K. Gummadi, R. Dunn, S. Gribble, and H. Levy, "An analysis of internet content delivery systems," *Oper. Syst. Rev.*, vol. 36, no. SI, pp. 315–327, 2002.

[25] A. Vakali and G. Pallis, "Content delivery networks: Status and trends," *IEEE Internet Comput.*, vol. 7, no. 6, pp. 68–74, 2003.

[26] C. Kalmanek, I. Ge, S. Lee, C. Lund, D. Pei, J. Seidel, J. van der Merwe, and J. Ates, "Darkstar: Using exploratory data mining to raise the bar on network reliability and performance," in *Proc. 7th DRCN*, 2009, pp. 1–10.

[27] R. R. Kompella, J. Yates, A. Greenberg, and A. C. Snoeren, "IP fault localization via risk modeling," in *Proc. 2nd NSDI*, 2005, pp. 57–70.

[28] R. Kompella, J. Yates, A. Greenberg, and A. Snoeren, "Detection and localization of network black holes," in *Proc. 26th IEEE INFOCOM*, 2007, pp. 2180–2188.

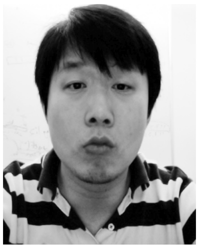
[29] I. Ilyas, V. Markl, P. Haas, P. Brown, and A. Aboulnaga, "Cords: automatic discovery of correlations and soft functional dependencies," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2004, pp. 647–658.

[30] S. Papadimitriou, J. Sun, and C. Faloutsos, "Streaming pattern discovery in multiple time-series," in *Proc. 31st Int. Conf. Very Large Data Bases*, 2005, pp. 697–708.

[31] K. Yamanishi and Y. Maruyama, "Dynamic syslog mining for network failure monitoring," in *Proc. 11th ACM SIGKDD KDD*, 2005, pp. 499–508.

[32] F. Le, S. Lee, T. Wong, H. Kim, D. Newcomb, F. Le, S. Lee, T. Wong, H. Kim, and D. Newcomb, "Minerals: Using data mining to detect router," in *Proc. ACM SIGCOMM MineNet*, 2006, pp. 293–298.

[33] J. Treinen and R. Thurimella, "A framework for the application of association rule mining in large intrusion detection infrastructures," *Lecture Notes Comput. Sci.*, vol. 4219, p. 1, 2006.



He Yan (M'12) received the M.S. degree in computer science from Colorado State University, Fort Collins, in 2009, and is currently pursuing the Ph.D. degree in computer science at Colorado State University.

He is a Senior Member of Technical Staff with AT&T Labs— Research, Florham Park, NJ. His current research interests are service quality management, network management and measurement, and Internet routing.

Lee Breslau received the M.S. and Ph.D. degrees in computer science from the University of Southern California, Los Angeles, in 1989 and 1995, respectively.

He is a Distinguished Member of Technical Staff with AT&T Labs, Florham Park, NJ. He was a member of the research staff with the Xerox Palo Alto Research Center, Palo Alto, CA, from 1994 to 1999 before joining AT&T. His research interests include Internet protocols, multicast routing, network measurement, and end-to-end service monitoring.

Zihui Ge (M'10) received the B.A. degree in computer science and technology from Tsinghua University, Beijing, China, in 1998, the M.S. degree in computer science from Boston University, Boston, MA, in 2000, and the Ph.D. degree in computer science from the University of Massachusetts, Amherst, in 2003.

Currently, he is a Principal Member of Technical Staff with the Networking and Services Research Center, AT&T Labs—Research, Florham Park, NJ. His research interests include IP network management, traffic analysis and anomaly detection, network data mining, and network security.

Dan Massey (SM'06) received the Ph.D. degree in computer science from the University of California, Los Angeles (UCLA), in 2000.

He is an Associate Professor with the Computer Science Department, Colorado State University, Fort Collins. He is currently the Principal Investigator on research projects investigating techniques for improving the Internet's naming and routing infrastructures. He is a co-editor of the DNSSEC standard (RFC 4033, 40334, and 4035). His research interests include protocol design and security for large-scale network infrastructures.

Dan Pei (SM'11) received the Bachelor's and Master's degrees from Tsinghua University, Beijing, China, in 1997 and 2000, respectively, and the Ph.D. degree from the University of California, Los Angeles (UCLA), in 2005, all in computer science.

He worked as a Principal Member of Technical Staff—Research with AT&T Labs—Research, Florham Park, NJ. His research interests include network management, measurement, and security.

Dr. Pei is a Senior Member of the Association for Computing Machinery (ACM).

Jennifer Yates (M'11) received the B.E. (hons) and B.Sc. degrees from the University of Western Australia, Crawley, Australia, in 1994, and the Ph.D. degree from the University of Melbourne, Melbourne, Australia, in 1998, all in computer science.

She is an Executive Director of Research with AT&T Labs—Research, Florham Park, NJ, leading the Network and Service Management Department. The department is inventing and prototyping future service and network management capabilities focused on mobility, IP and IPTV services, and driving these technologies to large-scale deployment across AT&T networks. Her earlier work focused on IP and optical network integration, ranging from network management and control, network reliability, IP control of optical networks (GMPLS), and IP and optical network integration. She was instrumental in AT&T's industry-leading optical mesh service deployment, which made the much-touted optical bandwidth on demand a commercial reality. Her recent research has focused on service quality management and advanced data mining to detect, troubleshoot, and resolve service and network issues.